Problem B. Bad Stack

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	512 мебибайт

Напомним, что обычный стек поддерживает две операции: добавление элемента и удаление элемента. При этом элемент, добавленный в стек последним, будет извлечен из стека первым.

Пусть теперь в стеке есть n дыр, каждая из которых находится на высоте h_i, а элемент в неё вываливается, если над ним находится минимум a_i элементов. Все элементы вываливаются одновременно (то есть если после добавления очередного элемента несколько элементов должны вывалиться в дыры, то они вывалятся все).

По описанию стека и списку операций выясните, сколько элементов будет безвозвратно утеряно через дыры в стеке.

Input

Первая строка входного файла содержит два целых числа n и $m(1 \le n \le 10^5,\ 1 \le m \le 3 \cdot 10^5)$ — количество дыр в стеке и количество операций, соответственно.

В последующих n строках находится по два числа — $1 \le a_i \le 3 \cdot 10^5$ и $1 \le h_i \le 3 \cdot 10^5$ — описание i-той дыры в стеке. Все числа a_i попарно различны.

В следующей строке содержится m символов `+` и `-`. Символ `+` обозначает операцию добавления элемента в стек, символ `-` — операцию удаления.

Output

В выходной файл выведите единственное число — количество элементов, вывалившихся через дыры.

Examples

standard input	standard output
2 4 1 1 2 1 +++-	2
2 5 1 2 2 1 +++++	4

Note

Рассмотрим первый пример:

1. Добавление. В стеке один элемент.
2. Добавление. В стеке два элемента. Один элемент вываливается в дыру на высоте 1 (поскольку над ним оказывается один элемент). Остается один элемент.
3. Добавление. Так же, как и на прошлом шаге, один элемент вываливается. Остается один элемент.

4. Удаление.

Таким образом, потеряется два элемента.

Рассмотрим второй пример:

1. Добавление. В стеке один элемент.
2. Добавление. В стеке два элемента. Элемент в дыру на высоте 1 не вываливается, поскольку над ним только один элемент, а в описании дыры указано, что требуется не менее двух.
3. Добавление. В стеке три элемента. Вываливается элемент в дыру на высоте 1 (потому что над ним оказывается два элемента). И вываливается элемент в дыру на высоте 2 (потому что над ним один элемент, а вываливаются элементы одновременно). В стеке остаётся один элемент.
4. Так же, как на шаге 2.
5. Так же, как на шаге 3.

Таким образом, потеряется четыре элемента.

Problem C. Cross

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 секунд
Memory limit:	512 мебибайт

Недавно байтландские антропологи опубликовали результаты исследования орнаментов одного из мультинезийских племён. Орнамент представляет собой поле $n \times m$, каждая клетка которого окрашена в белый или чёрный цвет и отличаются тем, что содержат ровно один максимальный правильный крест.

Определим *крест* размера $k \geq 1$ с центром в клетке (x, y) как фигуру, состоящую из горизонтальной полосы шириной в 1 клетку и длиной $2k + 1$ и вертикальной полосы шириной в 1 клетку и высотой $2k+1$, пересекающихся в клетке (x, y) так, что эта клетка является центральной для каждой из полос. Назовём крест *правильным*, если он раскрашен таким образом, что при повороте на 90 градусов вокруг центра чёрные клетки переходят в чёрные, а белые — в белые.

По заданному орнаменту определите максимальный размер и положение правильного креста, целиком в нём содержащегося.

Input

Первая строка входного файла содержит два целых числа n и m — количество строк и столбцов в орнаменте ($3 \leq n, m \leq 2000$). Каждая из последующих n строк содержит по m целых чисел от 0 до 1, разделённых пробелом. Нули обозначают белые клетки, единицы — чёрные. Гарантируется, что в орнаменте найдется ровно один крест максимального размера.

Output

Выведите три числа, разделенных пробелом — размер максимального правильного креста, номер строки и номер столбца, содержащего центр креста.

Example

standard input	standard output
5 5 1 1 0 1 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1	1 2 2

Problem D. Треугольники

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 мебибайт

На плоскости дано n чёрных и m белых точек. Найдите количество различных треугольников с вершинами в чёрных точках, не содержащих строго внутри себя белых точек. В частности, вырожденный треугольник не содержит строго внутри себя точек.

Input

В первой строке входного файла заданы два целых числа n и m ($0 \le n, m \le 500$) — количество чёрных и белых точек соответственно. i-я из последующих n строк содержат два целых числа x_i и y_i — координаты i-й чёрной точки. Аналогично, j-я из последующих m строк задаёт координаты j-й белой точки. Все координаты точек (x, y) удовлетворяют $-10^9 \le x, y \le 10^9$. Кроме того, все $n + m$ точек попарно различны.

Output

Выведите одно целое число — количество треугольников из чёрных точек, не содержащих строго внутри ни одной белой точки.

Example

standard input	standard output
4 1 7 0 3 7 3 4 0 0 2 1	2

Problem E. Поддеревья

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 секунды
Memory limit:	512 мебибайт

Дано дерево, вершины которого занумерованы числами от 1 до n, при этом корень дерева находится в вершине с номером 1.

Для каждой вершины v заданы два числа l_v и r_v.

Требуется отметить несколько вершин дерева так, чтобы для каждой вершины v в поддереве с корнем в v было отмечено не менее чем l_v и не более чем r_v вершин.

Гарантируется, что хотя бы один способ отметить вершины корректным образом существует.

Input

Первая строка входа содержит одно целое число n ($1 \le n \le 2 \cdot 10^5$) — количество вершин в дереве. i-я из следующих n строк описывает i-ю вершину. Сначала идут два целых числа l_i и r_i — наименьшее и наибольшее количество вершин поддерева с корнем в данной вершине, которое может быть отмечено. Далее идёт целое число p_i — количество потомков данной вершины. Далее перечисляются номера потомков.

Гарантируется, что номер каждой вершины, кроме корня дерева, встретится во входном файле ровно один раз.

Output

В первой строке выведите целое число m ($1 \le m \le n$) — количество отмеченных вершин. Во второй строке выведите m различных чисел — номера отмеченных вершин. Если ответов несколько, выведите любой из них.

Examples

standard input	standard output
1 1 1 0	1 1
5 2 2 2 2 3 0 2 2 4 5 0 0 0 1 1 0 0 1 0	2 2 4
4 4 4 1 2 0 3 1 3 0 2 1 4 0 1 0	4 1 2 3 4

Problem F. Game With Magic

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	512 мебибайт

Вдохновлённая успехом игры «1024 Stack Edition», байтландская компания 1D Games выпустила игру «Linear Puzzle Quest». В этой игре имеются n разноцветных шариков, расположенных на прямой. Игрок может выделить k или более подряд идущих одноцветных шариков, после чего эти шарики магически схлопываются вместе с занимаемым ими участком прямой (то есть шарики, соседние слева и справа с данным блоком, становятся соседними). Кроме того, игрок может вставить шарик любого цвета, заплатив одну монету. Цель игры — удалить все шарики.

По заданной позиции определите, какое минимальное число монет должен заплатить игрок, чтобы добиться цели

Input

Первая строка входного файла содержит два целых числа n и k — первоначальное количество шариков на прямой и минимальное количество подряд идущих шариков одного цвета, которые можно схлопнуть ($1 \le n \le 100$, $2 \le k \le 5$). Во второй строке заданы цвета шариков — n целых чисел c_i ($1 \le c_i \le 100$). Число c_i задаёт цвет i-го слева шарика в начальной расстановке.

Output

Выведите одно целое число — наименьшее количество монет, которое должен заплатить игрок, чтобы очистить поле.

Examples

standard input	standard output
2 4 3 3	2
10 4 1 1 1 1 3 1 2 2 2 1	4

Problem G. Byteland Union Bank

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 секунд
Memory limit:	512 мебибайт

Это интерактивная задача.

В базе старейшего байтландского банка Byteland Union Bank счета пронумерованы числами от 1 до n. На всех счетах открыта неограниченная кредитная линия (то есть сумма денег на счёте может быть и отрицательной). Используемая банковская система поддерживает три вида операций:

1. l r c — прибавить к счетам с номерами l, $l+1$, ..., r значение c.
2. d c — прибавить к счетам с номерами d, $2 \cdot d$, $3 \cdot d$ и так далее значение c.
3. l r — узнать суммарное количество денег на счетах l, $l+1$, ..., r.

Требуется проэмулировать работу данной системы.

Interaction protocol

Вначале программа жюри выводит количество счетов и начальные количества денег на них. Далее программа жюри выводит число, обозначающее количество операций, и сами операции. После каждой операции третьего типа программа участника должна вывести ответ на соответствующий запрос, и только после этого будет доступна информация о следующих операциях.

Input

Первая строка входа содержит одно целое число n ($1 \le n \le 10^5$) — количество счетов. Во второй строке находятся n целых чисел a_1, a_2, ... a_n ($|a_i| \le 10^9$) — начальное количество денег на каждом из счетов. В третьей строке находится целое число m — количество операций. Следующие m строк задают сами операции. Первое число — это тип операции (одно целое число от 1 до 3).

- Если тип операции равен 1, то за ним идут три целых числа l, r и c ($1 \le l \le r \le n$, $|c| \le 10^4$).
- Если тип операции равен 2, то за ним идут два целых числа d и c ($1 \le d \le n$, $|c| \le 10^4$).
- Если тип операции равен 3, то за ним идут два целых числа l и r ($1 \le l \le r \le n$).

Output

На каждую операцию третьего типа выведите в выходной поток сумму денег на указанных счетах. Не забудьте после вывода каждой строки использовать функцию `flush`.

Examples

standard input	standard output
10 1 2 3 4 5 6 -2 -3 -4 -5 9 3 4 8 1 7 10 2 3 7 10 2 2 -2 3 1 6 2 3 2 2 5 -7 1 2 8 4 3 3 10	 10 -6 15 20